CORBA/JMF 기반 오디오/비디오 스트림 시스템의 설계 및 구현

김만수† · 정목동††

요 약

분산 컴퓨팅 기술의 발전과 네트워크의 고성능화는 대용량의 멀티미디어 데이터를 송.수신 하여 처리 할 수 있게 되었다. 그러나 실시간 및 이기종간의 데이터 전송에는 네트워크 전송지연과 프로그램 구현의 어려움 등과 같이 극복해야 하는 문제점들이 많이 있다. 이 문제들을 해결하기 위하여 본 논문에서는 객체지향형 분산처리 환경 하에서 OMG가 제시한 멀티미디어 서비스 설계를 확장하는 방안을 제시한다. 이를 위해서 CORBA와 JMF Java Media API 기반으로 Smart Explorer라 불리는 효율적인 실시간 오디오/비디오 멀티미디어 소프트웨어 기반 구조를 제시하고 있다. 제어 데이터와 미디어 데이터의 전송 경로를 분리하고 효율적인 실시간 오디오/비디오 전송을 위해서 RTP/RTCP 프로토콜을 사용한다. 또한 제안하는 소프트웨어 기반 구조를 이용하여 오디오/비디오 스트림 시스템을 분산객체 환경에 맞게 구현한 결과도 제시한다. 향후 이 시스템은 방송, 원격 교육, 화상회의 등 실시간 통신 소프트웨어에 응용 될 수 있을 것으로 기대된다.

Design and Implementation of a CORBA/JMF-based Audio/Video Stream System

Mansoo Kim† and Mokdong Chung††

ABSTRACT

Recently, advances in high-speed networks and multimedia computer technologies allow new types of multimedia applications to manipulate large volumes of multimedia data. However, in the real time and/or the heterogeneous data transmissions, there are many difficulties such as network transmission delay, the implementation difficulties, and so on. To solve these problems, in this paper, we extend the method of the multimedia service design which is proposed by OMG. To do this, we suggest an efficient real time audio/video stream framework, called Smart Explorer, based on CORBA and JMF Java Media API. And we separate the transmission path of control data from that of media data and use RTP/RTCP protocol for efficient real time audio/video transmission. Also we show the appropriate implementation of the audio/video stream system based on our suggested framework Smart Explorer. In the future, we expect our audio/video stream system to be applied to the real time communication software such as broadcasting, distance learning, and video conferencing.

1. 서 론

최근 고속 네트워크와 멀티미디어 컴퓨터 기술의 발전과 더불어 일반 사용자들의 컴퓨팅 환경 또한 멀티미디어 서비스를 충분히 처리할 수 있는 수준으로 발전 하였다. 이러한 배경을 바탕으로 최근에는 화상회의나 주문형비디오시스템(Video On Demand : VOD), 인터넷 폰 등 새로운 형태의 분산멀티미디어 응용에 대한 연구가 활발히 진행되고 있다. 그러나 멀티미디어 서비스 기술의 미래에 대한 기대와는 달리 VoD, Voice over IP[1] 및 오디오/비디오 폰(Audio/Video Phone)과 같은 실시간을 요구하는 멀

† 준회원, 부경대학교 정보시스템협동과정 박사 과정
†† 정회원, 부경대학교 컴퓨터공학과 부교수

티미디어 응용은 이질적인 네트워크 환경이나 과중한 정보 전송으로 인한 네트워크 지연 등 아직 해결해야 할 많은 문제점들이 있으며, 이러한 멀티미디어 서비스에서 발생하는 문제점들을 극복하기 위해 많은 대안들이 연구되고 있다[2-4].

최근의 객체 기술은 소프트웨어 구성요소의 모듈화 및 재사용을 가능하게 할 뿐만 아니라, 컴퓨터 네트워크 상에서의 분산처리를 위한 기본 개념을 형성하는 데에도 큰 영향을 끼쳤다. 소프트웨어 업계를 중심으로, 객체 지향형 소프트웨어 모델링 기술을 분산처리 환경으로까지 확장하기 위한 연구가 계속되어, 이와 관련된 표준 사양들이 이미 정의되었거나 검토 중에 있다. 그 중에서도 OMG(Object Management Group)에서 분산 컴퓨팅과 객체지향 기술을 하나로 합친 표준 아키텍처로 제안된 CORBA(Common Object Request Broker Architecture)[5]는 분산시스템의 상호 운용성 문제를 해결하며, 객체지향적 방법을 제공함으로써 표준 개방 분산환경의 플랫폼으로 최근 주목을 받고 있다. 그러나 분산객체 미들웨어가 제공하는 많은 장점에도 불구하고 현재 대부분의 구현된 CORBA 프로그램은 텍스트 기반 중심으로써, 성능의 최적화, 실시간 기능 및 QoS(Quality of Service) 기능의 결여로 인하여 분산멀티미디어 응용에 효율적으로 적용되기 어렵다. OMG에서도 이러한 점을 인식하고 현재 오디오/비디오 스트림 처리에 관한 RFP(Request For Proposal)를 제시하고 있다 [6,7].

본 논문에서는 이와 같은 문제점들을 해결하기 위해 분산객체 미들웨어인 CORBA 환경을 확장하여 실시간 오디오/비디오 스트림의 전송과 제어를 효율적으로 처리하는 분산소프트웨어기반 Smart Explorer 프레임워크를 설계한다. 그리고 SUN사에서 제시한 다양한 플랫폼에서 구동 가능한 멀티미디어 애플리케이션 개발 Java Media API인 JMF(Java Media Framework) [14]를 사용하여 이를 응용한 오디오/비디오 스트림 시스템 Smile Phone을 구현 하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 연구 및 사례 등을 살펴보고 문제점들을 언급한 후, 본 논문의 배경을 간략히 소개한다. 3장에서는 Smart Explorer 프레임워크의 기본 요건들과 이에 근거하여 설계된 시스템의 전체 구성 및 세부 사항들에 대하여 기술하며, 4장에서는 3장의 설계에 기초하여 Smile Phone 시스템을 구현하고, 이를 시험한 내용들을 설명한다. 마지막으로 5장에서는 결론과 향후 연구과제에 관하여 기술한다.

2. 관련연구

컴퓨터 시스템들이, 각종 오디오/비디오 장치를 이용하는 응용소프트웨어를 운영하게 됨에 따라, 컴퓨터 네트워크 상에서 오디오/비디오 데이터를 효율적으로 처리하기 위한 연구들이 활발히 진행되고 있다. 또한 이질적인 네트워크 환경을 위한 분산객체 미들웨어가 이미 주요 소프트웨어 업체들에 의해 개발 되고 있다. 대표적으로 IONA의 Orbix, Microsoft DCOM, Inprise의 VisiBroker 등과 같은 여러 분산처리 미들웨어들은 이질적인 분산환경에서 위치의 투명성, 프로그래밍 언어의 투명성과 더불어 사용자에게 하드웨어와 소프트웨어 및 네트워크 환경의 일관성을 제공하게 됨에 따라 분산환경의 상호 운용성 문제를 자연스럽게 해결하고 있다[10-12].

CORBA는 분산객체들간에 서비스를 제공하거나 서비스를 획득할 수 있도록 하기 위한 통신 환경이 정의된 표준 기술이다. 그러나 CORBA는 텍스트 데이터 처리를 중심으로 설계 되어 있기 때문에 분산환경에서 멀티미디어 응용은 대용량의 멀티미디어 데이터 전송에 따른 과도한 통신 프로토콜의 오버헤드(Overhead), 시스템 내부처리의 오버헤드 및 실시간 서비스를 제공하지 못하는 근본적인 문제점이 있다. 이러한 이유로 OMG에서는 멀티미디어 스트림을 효율적으로 처리하기 위한 부가적인 표준을 정의 하였다. 현재 OMG에 제출되어 검토 단계에 있는 "오디오/비디오 제어 및 관리 사양"[7]은 CORBA기술과 오디오/비디오 데이터에 대한 처리 기술을 융합하기 위한 것이다. 이 제안서는 오디오/비디오 데이터 처리와 관련된 소프트웨어 구성 요소들을 CORBA 객체로 모델링하고, 이 CORBA 객체들간의 오디오/비디오 데이터 전송경로와 응용프로그램을 구성하는 객체들과의 제어정보 교환 경로를 분리하는 방안을 제시하고 있다. 그러나 이 제안은 분산응용프로그램을 작성하기에는 다소 복잡한 API를 정의하고 있으며, 아직 표준화 및 상용화 단계에는 이르지 못한 상태이다.

그림 1은 OMG의 오디오/비디오 스트림 서비스의 구조와 컴포넌트들을 보여준다. IDL 인터페이스들은 예를 들면, 스트림을 설정하고, 다운 받고, 전송을 시작하고, 정지하는 등의 스트림을 제어하고 관리하기 위해 사용되어 진다. 스트림은 연결을 위한 전송과 분리되어 운반되기 때문에 ORB 외부로 생각할 수 있다. 이것은 전화 통신에서 시그널링과 음성 채널이 분리되는 것과 어느 정도 유사하다.

본 논문에서 제시하는 분산멀티미디어 프레임워크인 Smart Explorer는 분산처리 환경에서 오디오/비디오 데이터를 처리하는 데 필요한 요소들을 수용하기 위한 기반구조를 설계 및 구현한 것이다. 또한 간단한 API만을 제공하여 쉽게 응용 시스템을 구축할 수 있으며, 응용시스템의 운영환경에 따라 소프트웨어 구성 요소들을 추가 혹은 제거할 수 있는 유연한 구조를 가지고 있다.

기존의 오디오/비디오 스트림 시스템들의 TCP/IP 및 UDP/IP를 이용한 일반적인 네트워크 프로그래밍을 통한 서버 또는 클라이언트간의 통신은 분산환경에서 이질적인 네트워크와 플랫폼과의 상호 연동성이 매우 어려워지고, 사용하는 프로토콜이 복잡해질수록 소프트웨어의 유지보수 비용이 증가하는 등 많은 문제가 발생하게 된다. 그러나 CORBA를 기반으로 설계 및 구현된 오디오/비디오 스트림 프레임워크 Smart Explorer는 IDL(Interface Definition Language)을 통해 객체 메소드들을 규정하여 시스템 연결 및 스트림 전송제어를 운영시스템 및 네트워크에 투명하도록 함으로써 상호 연동성 문제를 쉽게 해결하고, 멀티미디어 데이터의 실시간 캡쳐, 전송, 재생 등을 Java기반 JMF API로 구현하여 기존 오디오/비디오의 단점인 플랫폼 종속성을 제거

그림 1. OMG에서 Stream Service를 위해 제시한 시스템 구성도

한다. 또한 객체지향 설계 및 구현으로 프로그램의 복잡성이 현저하게 줄어 소프트웨어의 수정이나 기능 추가 등 유지보수 비용을 최소화 한다.

3. 시스템 설계

실시간 오디오/비디오 데이터 처리를 위한 Smart Explorer 프레임워크는 분산객체 환경과 함께 미들웨어를 형성하는 부분으로써 미들웨어가 요구하는 요건들을 만족하여야 한다. 즉, 응용 프로그램들에게 잘 정의된 API를 제공함과 동시에, 하부 운영체제가 제공하는 시스템 자원들을 효율적으로 이용하도록 설계되어야 한다.

3.1 설계 목표

본 연구는 다음과 같은 설계 목표를 가진다.

① 상호운용성(interoperability) 제공

이질적인 분산환경에서 분산객체 미들웨어 CORBA에서 제공하는 상호 연동성을 자연스럽게 지원받는다.

② 제어 데이터와 스트림 데이터의 전송 분리

시간에 민감하고 대용량인 스트림 데이터는 미들웨어를 통하지 않고 하부 네트워크 프로토콜로 바로 전송하여 스트림 전송의 효율을 높인다.

③ 플랫폼 독립성 제공

Java와 Java Media API인 JMF로 시스템을 구현함으로써 운영 플랫폼에 투명성을 제공한다.

④ 다양한 미디어 형식 지원

MPEG-I, II, MJPEG, G.721, G.723.1 등 다양한 미디어 코덱(Codec)을 지원한다.

⑤ 객체지향설계

UML(Unified Modeling Language)[12]을 사용한 객체지향 설계와 구현 언어를 사용하여 소프트웨어 생산성을 높인다.

3.2 전체 시스템의 구조

본 논문에서는 OMG에서 제시한 CORBA 기반 멀티미디어 스트림 서비스 설계를 기초로 OMG의 복잡한 API를 단순화하여 쉽게 응용시스템에 적용할 수 있도록 한다. 이를 위하여 효율적인 오디오/비디오 데이터 처리를 위한 일반적인 스트림 제어용 분산환경 소프트웨어 설계와 Java기반 클라이언트 프레

임워크인 Smart Explorer을 설계하고, 이를 응용한 오디오/비디오 스트림 시스템 Smile Phone를 구현한다.

그림 2는 Smart Explorer 프레임워크의 전체 분산객체 시스템 구조를 보여 주고 있다. 그림 2에서와 같이 연결 및 여러 멀티미디어 전송 제어 및 관리는 IDL로 정의된 분산객체들에 의해 처리되어 ORB를 통해 통신하고, 실시간 오디오/비디오 멀티미디어 데이터는 JMF가 제공하는 각 클라이언트의 디바이스 핸들러(Device Handler)에서 처리하여 RTP(Real-Time Transport Protocol)[8,9]로 전송 된다.

한정된 대역폭의 네트워크 환경은 실시간 멀티미디어 데이터 전송 과정에 데이터의 손실 및 지연이 발생할 수 있고, 이러한 손실 및 지연 정보는 RTP 헤더 정보의 순번(Serial Number), 지터(Jitter), 왕복시간(Round-Trip Times)을 분석해 알 수 있다. 현재 이와 같은 문제점을 해결하고 네트워크의 대역폭에 따른 손실을 최소화 하기 위해 오디오 정보의 우선 전송, 낮은 대역폭으로의 코덱 변환 등을 적용하고 있다.

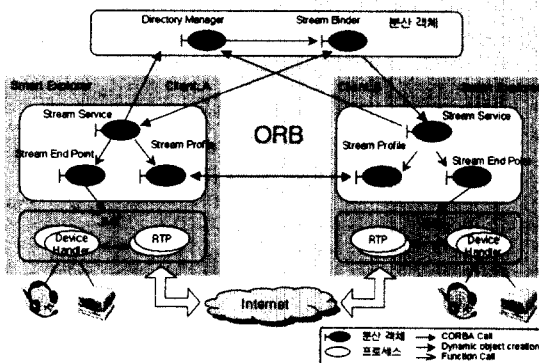



그림 2. Smart Explorer 프레임워크 전체 시스템 구조

3.3 Smart Explorer 프레임워크 분산객체들

그림 3은 Smart Explorer 프레임워크의 오디오/비디오 멀티미디어 데이터 전송을 위한 분산객체들간의 관계 설정을 나타낸다. 본 시스템에서는 DirectoryManager, StreamBinder, StreamService, StreamEndPoint, StreamProfile 분산객체들이 오디오/디오 데이터 관리 및 전송 제어를 담당한다. DirectoryManager와 StreamBinder는 분산시스템에서 클

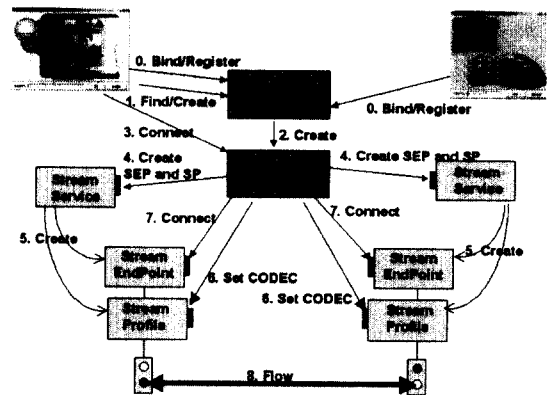



그림 3. Smart Explorer의 스트림 전송을 위한 분산객체 관계

라이언트로부터 연결을 위해 대기한다. StreamService, StreamEndPoint, StreamProfile은 각 클라이언트에서 생성되며 StreamBinder를 통해 클라이언트 상호간에 객체 참조를 하게 된다.

① DirectoryManager

DirectoryManager 객체는 등록된 StreamService 객체로부터 클라이언트 연결 정보를 DB 또는 리스트로 저장하고, 연결을 희망하는 클라이언트의 요구에 의해 등록된 다른 StreamService 리스트 정보를 제공하게 된다. 그리고 두 클라이언트간의 연결을 위해 StreamBinder 객체를 생성한다.

```
interface  DirectoryManager
{
           boolean    Register(in string strPhoneNum,
                           in StreamService ssObj);
           boolean    Unregister(in string strPhoneNum);

           StreamService Find(in string strPhoneNum);

           boolean    FindFirst();
           boolean    FindNext();
           boolean    GetAt(out FindInfo      fInfo);

           StreamBinder CreateSB();
           boolean DestoryCM(in StreamBinder cm);
};
```

② StreamBinder

StreamBinder는 연결을 희망하는 두 클라이언트의 StreamService 객체의 연결을 관리한다. StreamBinder 객체는 DirectoryManager로부터 생성되며, 하나의 연결 당 하나의 StreamBinder가 존재한다.

```
interface StreamBinder
{
    oneway void    Connect(in StreamService ssObj_A,
                           in StreamService ssObj_B);
    oneway void    Disconnect();

    boolean   Play();
    boolean   Stop();
};
```

③ StreamService

StreamService 객체는 StreamEndPoint, StreamProfile 객체를 생성한다. StreamBinder에 의해 StreamService 객체간 연결이 이루어지며, 연결시점에서 두 StreamProfile 객체간의 지원 멀티미디어 데이터 정보에 따라 연결 여부를 확인 후 연결 과정을 완료한다.

```
interface StreamService
{
        readonly attribute string PhoneNum;

        readonly attribute PhoneState State;

        StreamEndPoint        CreateSEP();
        StreamProfile         CreateSP();
};
```

④ StreamEndPoint

StreamEndPoint는 멀티미디어 데이터 전송 제어를 관리하게 된다. StreamEndPoint는 오디오/비디오 데이터 처리를 위한 AudioManager, VideoManager 객체에 실시간 데이터 캡쳐, 재생과 네트워크를 통한 데이터 전송을 제어한다. 데이터의 전송 상태는 AudioManager, VideoManager 객체의 RTP 정보를 얻어 네트워크 상태 및 시스템 상태 등을 주기적으로 확인함으로써 원활한 전송을 보장하게 한다.

```
interface StreamEndPoint
{
        readonly attribute string RTP_IP;
        readonly attribute long RTP_Port;

        boolean Connect(in StreamEndPoint sep_Dest,
                  in string DestIP, in long DestPort);
        boolean Disconnect();

        boolean    Play();
        boolean    Stop();
};
```

⑤ StreamProfile

StreamProfile 객체는 StreamService 객체가 지원하고자 하는 미디어의 형식을 정의하고 있으며, StreamBinder에 의해 StreamService 객체의 연결이 이루어지는 시점에서 지원하는 미디어에 대한 상호 지원 여부를 확인하게 된다.

```
interface StreamProfile
{
boolea SupportAudioCodec(in AudioCodec chkAudioCodec);
boolea SupportVideoCodec(in VideoCodec chkVideoCodec);
};
```

3.4 Smart Explorer 프레임워크 구조

그림 4는 Smart Explorer 프레임워크의 컴포넌트 다이어그램을 나타내고 있다. Smart Explorer 프레임워크의 확장성 및 이식성을 고려하여 통신 모듈 ConnectionManager와 멀티미디어 모듈인 VideoManager, AudioManager로 구성 된다.

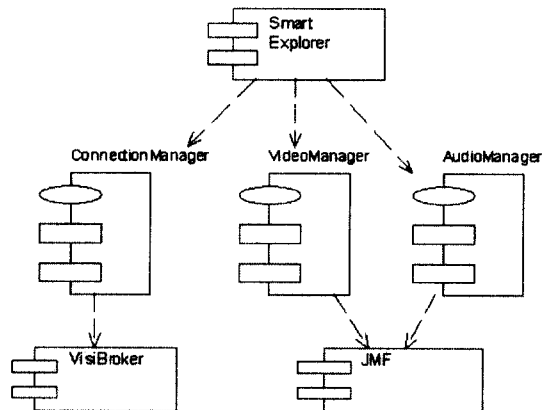



그림 4. Component View

그림 5는 Smart Explorer 프레임워크의 내부 구조이다. 응용 애플리케이션들은 Smart Explorer 프레임워크 API를 사용하여 연결 및 데이터 송,수신을 하게 된다. Smart Explorer 프레임워크는 StreamService, StreamEndPoint, StreamProfile 분산객체들을 생성, 다른 클라이언트와 연결하는 ConnectionManager와 오디오/비디오 데이터 송,수신을 위해 VideoManager와 AudioManager 객체를 생성한다. 효율적인 스트림 전송을 위해 비디오 코덱은 JPEG을 사용하고, 오디오 코덱은 G.723을 사용한다. 비디오/오디오 코덱은 JMF에서 제공하는 모든 코덱을 제공함으로써 유연성을 제공한다.

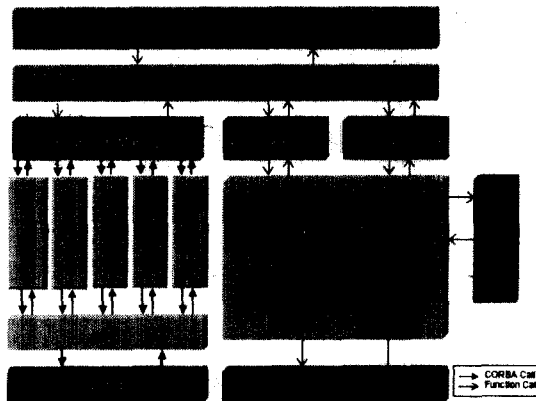



그림 5. Smart Explorer 프레임워크의 내부 구조

Smart Explorer 프레임워크의 한 부분인 ConnectionManager 모듈은 분산객체 간의 통신을 담당하는 클래스로 StreamService, StreamEndPoint, StreamProfile 객체들의 생성, 등록, 검색, 연결 등의 기능을 수행한다. ConnectionManager 모듈은 그림 3에서 보여지는 다른 시스템간의 분산 스트림 객체간 통신을 Smart Explorer 프레임워크 API에 의해 ConnectionManager와 SmartExplorer 클래스간 프레임워크 내부에서 이들에 대한 관리를 함으로써 애플리케이션에 CORBA 객체의 상호간 제어에 대해 투명성을 제공한다.

VideoManager와 AudioManager 모듈은 각각 Video와 Audio의 화상 및 음성 캡쳐와 재생의 모든 기능이 구현된 클래스이다. 이들 클래스는 JMF에서 제공하는 캡처, 재생용 데이터소스(DataSource)와 RTP 수신, 전송용 데이터소스를 생성하고, 이 데이터소스 코덱을 담당하는 프로세서(Processor)와 연결시킨다. 그리고, 역시 JMF에서 제공하는 SessionManager로 RTP 수신, 전송 데이터소스와 캡처, 재생 데이터소스를 연결하여 스트림 데이터의 전송을 제어한다.

SmartExplorer 클래스는 ConnectionManager, VideoManager, AudioManager 클래스들의 이벤트 처리를 위하여 Java로 정의된 인터페이스를 각각 구현함으로써, 각 모듈의 제어 및 모듈간의 메시지를 중계한다.

표 1은 Smart Explorer 프레임워크 API를 보여준다. API는 오디오/비디오 시스템 구현에 필요한 간단한 API를 제공한다.

그림 6은 Smart Explore 프레임워크 간의 통화 시나리오를 UML 시퀀스 다이어그램(Sequence Diagram)으로 표시 한 것이다. 클라이언트 A와 B의 응용 애플리케이션들은 분산객체와 오디오/비디오 전송을 위한 SmartExplorer 객체를 생성하여 다른 클라이언트와의 연결을 준비한다. SmartExplorer는 생성한 StreamService 객체를 DirectoryManager에 등록한다. 클라이언트 A에서 클라이언트 B에 연결을 위해 먼저 클라이언트 A는 DirectoryManager로부터 클라이언트 B의 StreamService 객체를 찾는다. 객체를 찾기 위해서는 그 객체의 등록된 ID나 IP 등 객체를 식별할 수 있는 정보를 이용한다. 클라이언트

표 1. Smart Explorer 프레임워크 API

함 수 이 름	설 명
boolean CreateSS(String strMyPhoneNum)	StreamService 객체 생성
boolean DestorySS()	생성된 StreamService 객체 소멸
boolean Register(String strDM_URL)	StreamService 객체를 DirectoryManager에 연결
boolean Unregister()	DirectoryManager에 연결 해제
boolean Find(String strPhoneNum)	통화 상대자 검색
boolean Connect(strPhoneNum)	통화 상대자와 연결
boolean Disconnect()	현재 통화 연결 해제
boolean Start()	스트림 전송 시작
boolean Stop()	스트림 전송 중지
boolean CreateVideoDevice(VideoFormat vf)	비디오 디바이스 객체 생성
boolean DestoryVideoDevice()	비디오 디바이스 객체 소멸
boolean CreateAudioDevice(AudioFormat af)	오디오 디바이스 객체 생성
boolean DestoryAudioDevice()	오디오 디바이스 객체 소멸
boolean SetVideoPlayerPanel(Panel p)	비디오 재생 패널 설정
boolean SetAudioPlayerPanel(Panel p)	오디오 재생 패널 설정

A는 자신의 StreamService의 객체와 검색된 클라이언트 B의 StreamService 객체로부터 Stream-EndPoint와 StreamProfile 객체를 생성하게 하고, 두 클라이언트의 StreamProfile 객체를 상호 주고 받음으로써 연결을 위한 오디오/비디오 코덱을 설정한다. StreamProfile 객체에 의해 상호간 오디오/비디오 코덱이 설정되면 클라이언트 A와 클라이언트 B의 StreamEndPoint의 play() 메소드를 호출하여 오디오/비디오 전송을 시작한다.

그림 6. 통화 시나리오 시퀀스 다이어그램

4. 구현 및 테스트

Smile Phone은 분산멀티미디어 프레임워크인 Smart Explorer를 응용한 화상시스템 애플리케이션이다. Smile Phone은 화상카메라와 마이크를 장착한 Windows 95, 98/NT/2000을 운영체제로 사용하는 펜티엄(Pentium)급 PC들을 대상으로 구현되었다. 분산객체 환경을 구성하기 위해서 서버에 Inprise사의 VisiBroker 4.5 버전을 설치 하여 Stream Service CORBA객체를 구현 하였다. 클라이언트는 Windows 98 환경에서 JDK 1.3을 지원하는 J-Builder 4.0과 오디오/비디오 처리를 위해 JMF를 이용하여 구현하였다.

그림 7은 DirectoryManager로부터 받은 현재 연결된 사용자의 정보를 표시한 것이다. Directory-Manager에서는 사용자의 이름과 IP 정보가 표시 되고, 통화를 원하는 사용자를 더블클릭하면 상호간 통화를 위한 연결 설정이 이루어진다.

통화 연결이 이루어지면 네트워크 상태를 실시간으로 표시하는 화면이 그림 8처럼 보여진다. 네트워크 상태는 RTP로부터 데이터를 받은 정보를 통해

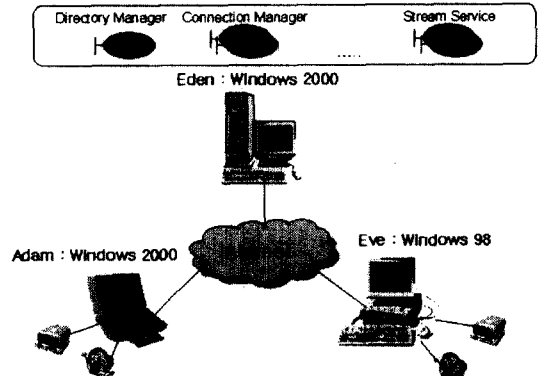



그림 7. 시스템 테스트 환경

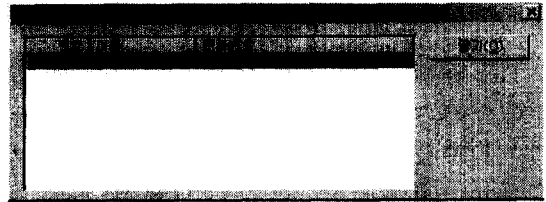



그림 8. 디렉토리 서비스 화면

패킷(Packet) 손실 여부를 확인하고 일정 손실 정도를 계산하여 화면에 반영한다.

그림 9에서 보는 것처럼 상대방주소 즉, 연결하고자 하는 다른 클라이언트의 StreamService 객체가 DirectoryManager에 등록한 상대방 IP를 입력하여 연결버튼을 누르면, 3장의 설명과 같이 Directory-Manager에서 입력한 IP에 맞는 StreamService객체를 검색하여 오디오/비디오 스트림 제어를 위한 StreamEndPoint 객체를 생성하고, 두 클라이언트간

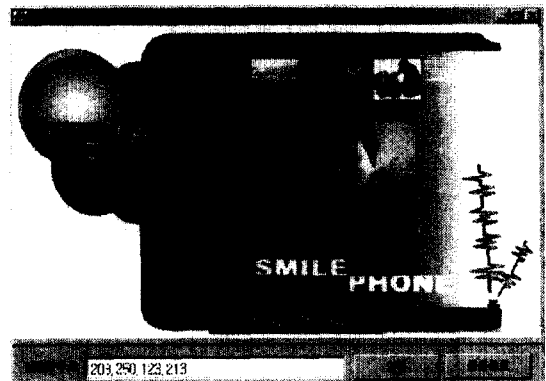



그림 9. 통화 화면

에 PC-to-PC 연결하여 오디오/비디오 데이터를 전송하게 한다. 이렇게 구현 함으로써 연결 및 오디오/비디오 스트림 제어 및 관리와 오디오/비디오 스트림 전송을 분리하였고, CORBA와 Java를 기반으로 함으로써 플랫폼에 상관없이 동작하게 된다.

일반적인 LAN을 통한 인터넷 접속을 하여 두 클라이언트의 연결시 발생되는 오디오 데이터의 손실과 56Kbps 모뎀 PPP 접속을 하여 발생되는 오디오 데이터의 손실은 56Kbps 모뎀 PPP 접속시 오디오 데이터의 손실이 더 많다는 것을 실험을 통해 확인 할 수 있었다. 이러한 오디오 데이터의 손실을 최소화 하기 위해 비디오 데이터 보다 오디오 데이터의 우선 전송이나 오디오 코덱의 실시간 변경에 대해서는 지속적인 연구가 필요하다.

Java 애플리케이션으로 개발된 Smile Phone은 동일한 프로그램으로 Windows 98이나 Windows 2000 시스템인 PC환경 뿐만 아니라 Unix나 Linux에서도 수행 될 수 있는 장점이 있다. 뿐만 아니라, 직관적인 인터페이스를 제공함으로써 사용하기에 편리하다.

일반적으로 RTP를 사용하지 않은 CORBA기반 멀티미디어 스트림 시스템들은 비록 분산객체들의 통신만으로 좀더 직관적인 API를 제공할 수 있는 장점이 있지만, 대용량의 실시간 데이터를 전송하는 것에는 아직 네트워크 인프라가 구축이 되어 있지 못한 실정이다. Smile Phone은 이러한 점에서는 효율적인 데이터 전송을 보장하지만, 오디오/비디오 코덱 및 전송을 위해 JMF로 구현 함으로써 사용자가 본 시스템을 사용하기 위해서는 JMF를 설치하는 번거로움이 있으며, 아직 Java의 처리 속도에 따른 높은 시스템 사양이 요구되는 단점이 있다.

5. 결론 및 향후 과제

분산 컴퓨팅 기술의 발전과 네트워크의 고성능화는 대용량의 멀티미디어 데이터를 송.수신 하여 처리 할 수 있게 되었다. 그러나 실시간 및 이기종간의 데이터 전송에는 네트워크 전송지연과 프로그램 구현의 어려움 등과 같이 극복해야 하는 문제점들이 많이 있다. 본 논문에서는 이와 같은 문제점들을 해결하기 위해 분산객체 미들웨어인 CORBA 환경을 확장하여 실시간 오디오/비디오 스트림의 전송과 제어를 효율적으로 하기 위한 분산소프트웨어기반 구조 Smart Explorer 프레임워크를 설계하고, SUN사에서 제시한 다양한 환경의 플랫폼에서 구동 가능한 멀티미디어 애플리케이션 개발 Java Media API인 JMF를 사용하여 이를 응용한 오디오/비디오 스트림 시스템 Smile Phone을 구현 하였다.

Smart Explorer 프레임워크는 분산 스트림 객체를 사용하여 시스템간의 연결, 스트림 제어, 관리를 효과적으로 처리하고 있고, 텍스트 기반인 CORBA의 실시간 멀티미디어 데이터 처리에 따른 문제점을 보완하기 위해 멀티미디어 스트리밍 처리를 JMF를 통해 오디오/비디오 스트림 형식의 실시간 데이터를 체계적으로 처리하였다. 또한 Smart Explorer 프레임워크를 응용한 오디오/비디오 스트림 시스템인 Smile Phone을 구현하여 Smart Explorer 프레임워크의 효율성을 확인하여 보았다. 실험 결과 분산객체간의 연결 및 제어와 실시간 멀티미디어 데이터의 원활한 전송을 확인할 수 있었다.

그러나 제안한 프레임워크 Smart Explorer는 QoS를 구현하지 못했기 때문에 네트워크의 대역폭의 변화에 민감하게 반응을 보임으로써, 효과적인 전송을 보장하지 못할 수도 있다. 이를 극복하기 위해서는 QoS 분산객체를 설계하고, 이 객체와 스트림 객체간의 상호 정보 전달을 통한 문제 해결이 필요하며, 본 연구 결과는 향후 인터넷상의 방송, 교육, 화상회의 및 모바일 환경에서 실시간 멀티미디어 스트리밍 분야에 응용할 수 있다.

참 고 문 헌

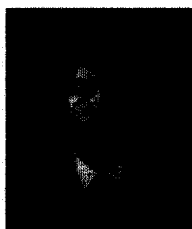
[1] D. Minoli and E. Minoli. Delivering Voice Over IP Networks. John Wiley, 1998.

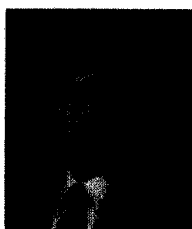
[2] Guojun Lu, Communication and Computing for Distributed Multimedia Systems, Artech House, Boston. London, 1996

[3] T.H. Yun, J, Y. Kong and J. Won-Ki Hong, A CORBA-based Distributed Multimedia System, Proc. of 1997 Pacific Workshop on Distributed Multimedia Systems, pp. 1-8, 1997

[4] 이성환, CORBA에 기반한 DSM-CC 구현, KRNET '97, pp.505-517, 1997

[5] OMG, The Common Object Request Broker: Architecture and Specification Revision 2.4,

OMG Document formal , October 2000
[6] OMG, Control and Management of A/V Streams Request For Proposal, OMG TC Document, 1996
[7] OMG, Audio/Video Streams v1.0, OMG Document formal, 2000
[8] RFC 1889 RTP: A Transport Protocol for Real-Time Applications, January 1996
[9] RFC 1890 RTCP : RTP Profile for Audio and Video Conferences with Minimal Control, January 1996
[10] Jon Siegel, CORBA Fundamental and Programming, John Wieley & Sons, Inc. , 1996
[11] IONA Technologies PLC, Orbix Programmer's Guide, IONA Technology Ldt., 1997
[12] Inprise Corporation, Programmer's Guide. VisiBroker for Java, 2000
[13] Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Addion-Wesley, 1998
[14] Inc. Sun Soft. Java Media Framework, http://ja-va.sun.com/products/javamedia/jmf/, 1998

김 만 수

1996년 동서대학교 컴퓨터공학과 학사
2001년 부경대학교 컴퓨터공학과 석사
2001년 3월~현재 부경대학교 정보시스템협동과정 박사과정
관심분야 : Intelligent Agent, Electronic Commerce, Bluetooth, Internet Phone

정 목 동

1981년 경북대학교 컴퓨터공학과 학사
1983년 서울대학교 컴퓨터공학과 석사
1990년 서울대학교 컴퓨터공학과 박사
1984년~1985년 금성반도체(주) 연구소 연구원
1985년~1996년 부산외국어대학교 컴퓨터공학과, 부교수
1996년~현재 부경대학교 컴퓨터공학과, 부교수
1999년~2000년 미국 Iowa State University, Visiting Professor
관심분야 : Intelligent Agent, Electronic Commerce, Distance Learning